Capacitor :
from PWA to native app in 5 minutes
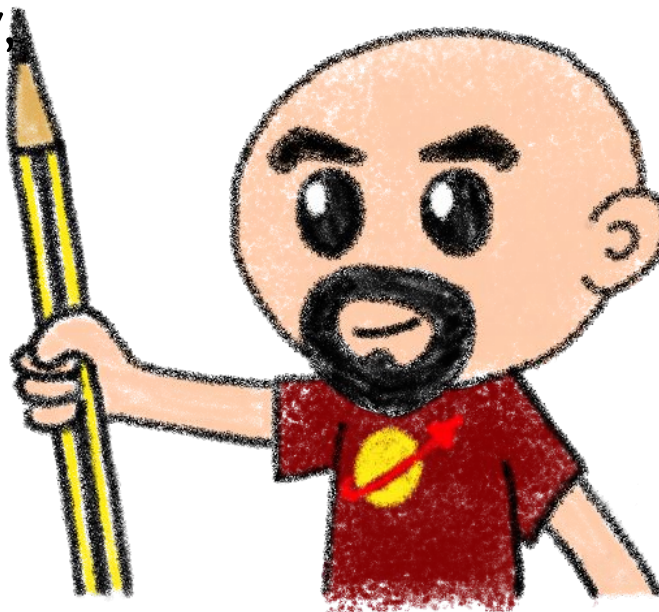
Horacio Gonzalez @LostInBrittany

Who am I?

Introducing myself and introducing OVH

Horacio Gonzalez

@LostInBrittany

Spaniard lost in Brittany,
developer, dreamer and
all-around geek

OVH
Team DevRel

OVH : Key Figures

1.3M **Customers worldwide in** **138** **Countries**

1.5 **Billions euros** **investment over five years**

30 **Datacenters (growing)**

350k **Dedicated Servers**

200k **Private cloud VMs running**

650k **Public cloud Instances created in a month**

15TB **bandwidth capacity**

35 **Points of presence**

4TB **Anti DDoS capacity**

Hosting capacity : **1.3M** **Physical Servers**

+ 2 500 **Employees in** **19** **countries**

20 **Years of Innovation**

Ranking & Recognition

1st **European** Cloud Provider*

1st **Hosting** provider in Europe
1st **Provider** Microsoft Exchange
Certified vCloud Datacenter
Certified Kubernetes platform (CNCF)
Vmware **Global Service Provider** 2013-2016
Veeam Best Cloud Partner of the year (2018)

* *Netcraft 2017 -*

OVH: Our solutions

Cloud

VPS
Public Cloud
Private Cloud
Serveur dédié
Cloud Desktop
Hybrid Cloud

Mobile Hosting

Containers
Compute
Database
Object Storage
Securities
Messaging

Web Hosting

Domain names
Email
CDN
Web hosting
MS Office
MS solutions

Telecom

VoIP
SMS/Fax
Virtual desktop
Cloud HubiC
Over theBox

Capacitor

The Native Bridge for Cross-Platform Web Apps

What's Capacitor?
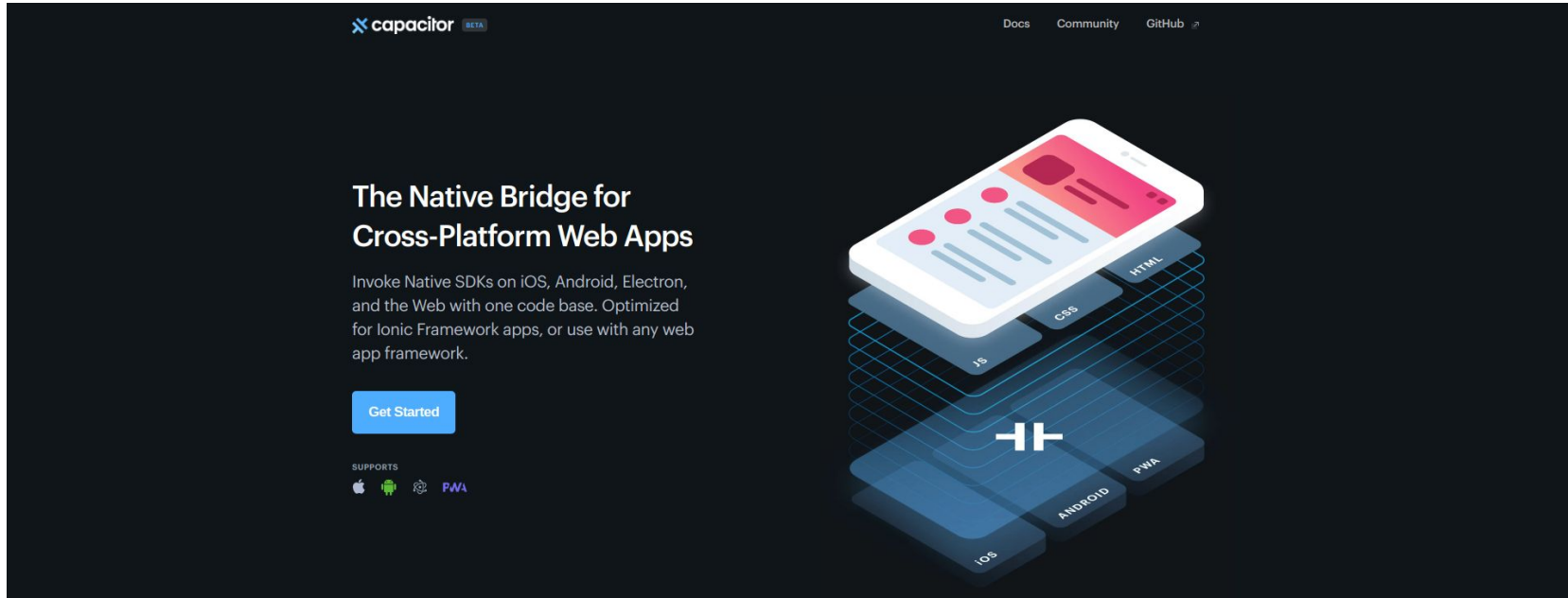
Cross-platform app runtime making it easy to build web apps that run natively on iOS, Android and web

Spiritual heir to Apache Cordova

Evolution, not revolution

Spiritual heir to Apache Cordova

Support for many Cordova plugins

Extensible and evolutif

- Close to web-standards

- Plugin API
  - Swift on iOS
  - Java on Android
  - JavaScript for the web

Developer Friendly

Easy to get started

Works on any framework

You still need the platform tools

Android Studio and/or Xcode
to build the native packages

Weren't you a PWA advocate?

And you are also championing Flutter!
Where is the coherence, guy?

Well, I am a PWA advocate indeed

I rooted for PWA before it was fancy…

And I know the numbers...

Top 1000 mobile apps vs top 1000 mobile web properties

■ Web mobile   ■ Apps

8.9

3.3

201.8

10.9

Monthly unique visitors (millions)
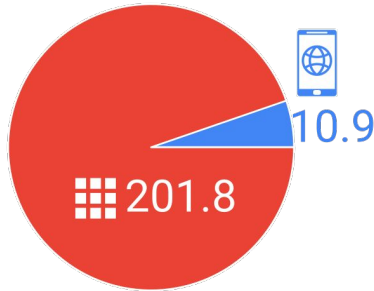
Average minutes by visitor

Source: comScore Mobile Metrix 2015, U.S., Age 18

Apps drive engagement, web drive visitors...

Millennials' Top Apps by Share of Total Mobile App Time Spent
Source: comScore Mobile Metrix, U.S., Age 18+, June 2015

41.5%

21.2%

14.3%

5.6%

3.9%

3.2%

3.1%

2.6%

1.6%

1.5%

1.5%

■ Facebook
■ Pandora Radio
■ YouTube
■ Instagram
■ Spotify
■ Facebook Messenger
■ Snapchat
■ SoundCloud
■ Netflix
■ Google Search
■ All Other
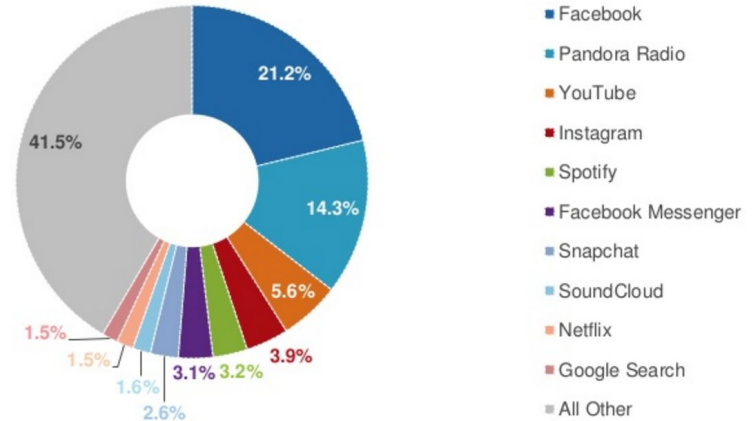
20 biggest apps account for 80% of user time

Saison 3
DevFest
LILLE

An engineer role is to choose

The right tool for the right problem

If you need super fancy UX

60 FPS, slick animations
An experience more than an app

Then go **full native**

And BTW, do it in **Flutter**

believe me, you will thank me!
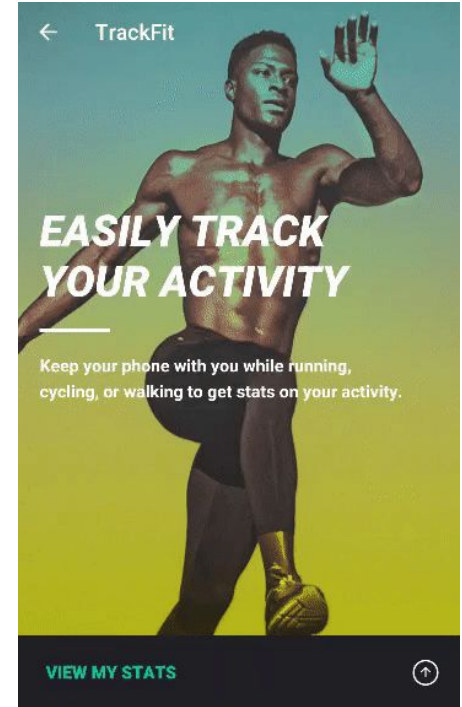
For more normal needs

The Washington Post

AMP helps the Washington Post increase returning users from mobile search by 23%!

"We are committed to improving speed across the board. If our site takes a long time to load, it doesn't matter how great our journalism is, some people will leave the page before they see what's there"
David Merrell, Senior Product Manager, The Washington Post

READ MORE

AliExpress

"Smarter shopping, better living!" is the motto of AliExpress, a website where shoppers can buy everything from baby clothes to refrigerators directly from China. Part of the Alibaba Group, the global online retail marketplace is now a popular e-commerce site in America, Russia, and Brazil.

READ MORE
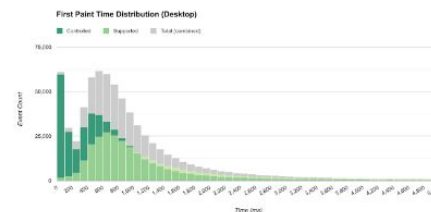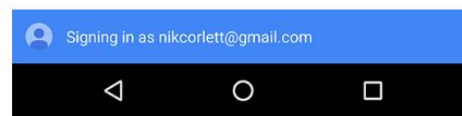
Measuring the Real-world Performance Impact of Service Workers

One of the most significant benefits of service workers (from a performance perspective, at least) is their ability to proactively control the caching of assets. A web application that can cache all of its necessary resources should load substantially faster for returning visitors. But what do these gains actually look like to real users? And how do you even measure this?

READ MORE

A well done PWA is simply enough

But if you need to be in the store?

For many reasons, not all objective...

Hybrid PWA apps

The best of two worlds

Capacitor take your PWA to the store
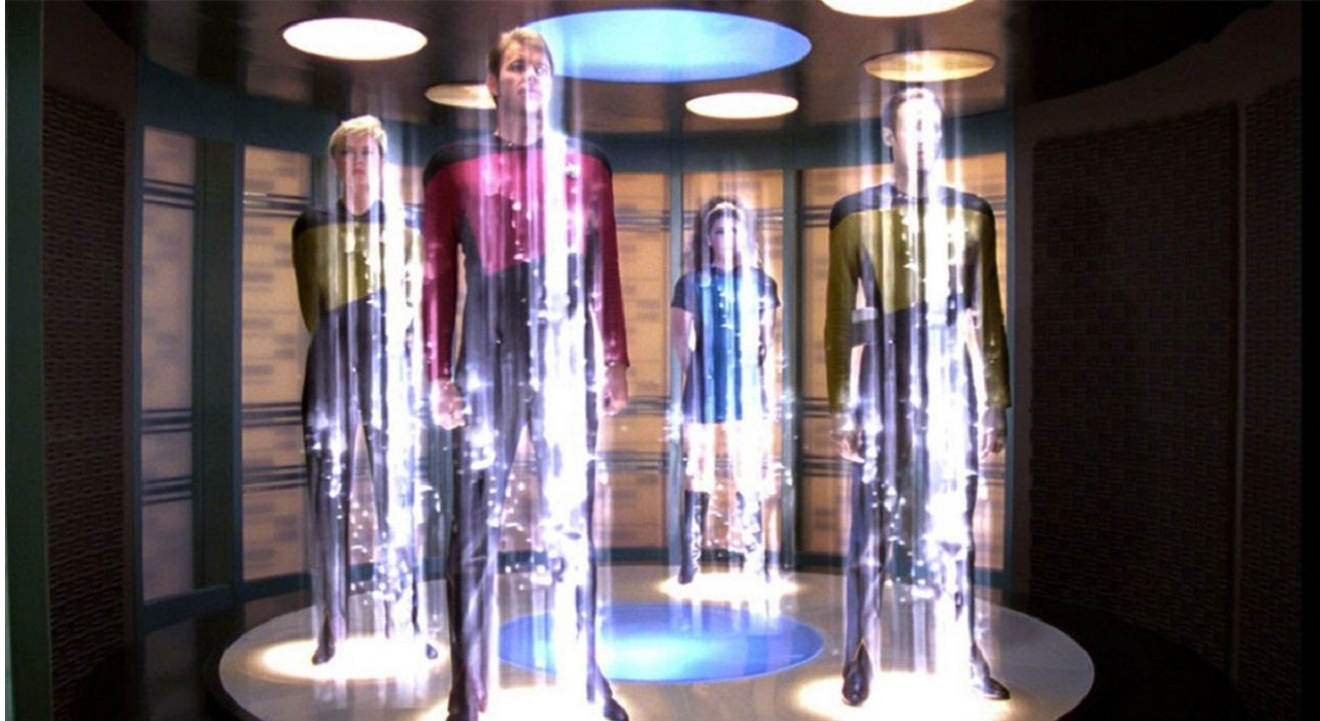
In a simple, quick and painless way

First steps testing Capacitor

Adding Capacitor to an existing app

From web to native

Giving superpowers to you webapp

Testing with a real webapp

Warp 10 Photon - IDE for Warp 10

Step 1 - Add Capacitor to the app

- Install Capacitor

```
cd my-app

npm install --save @capacitor/core @capacitor/cli
```

- Init Capacitor

```
npx cap init
```

- Add Android and/or iOS and/or Electron support

```
npx cap add android
```

Step 2 - Copy to Android

- Edit `capacitor.config.json` to choose the built folder

- Copy the built resources to Android

```
npx cap copy
```

- Launch Android Studio

```
npx cap open android
```

Step 3 - Test
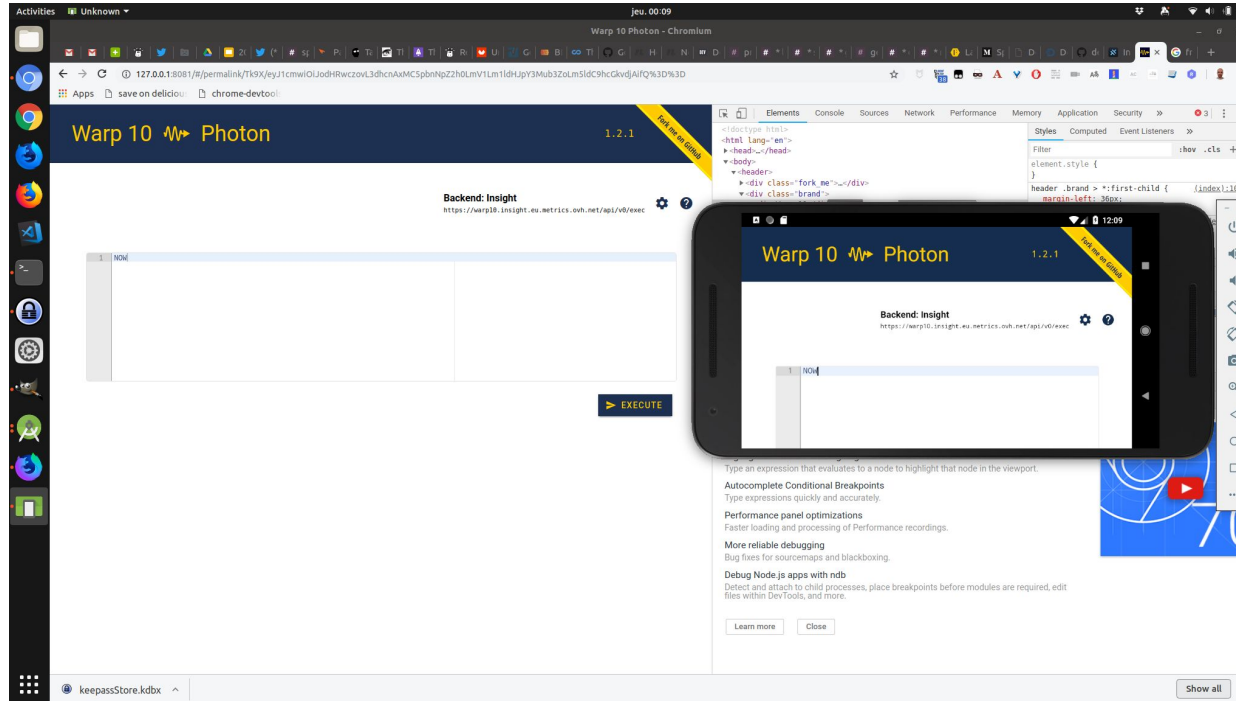
And our webapp is now a native app

First test: successful!

Capacitor 1 - Scepticism 0

The Capacitor example app

Nice to explore Capacitor

Made by the Ionic team

To replace Cordova in Ionic 4
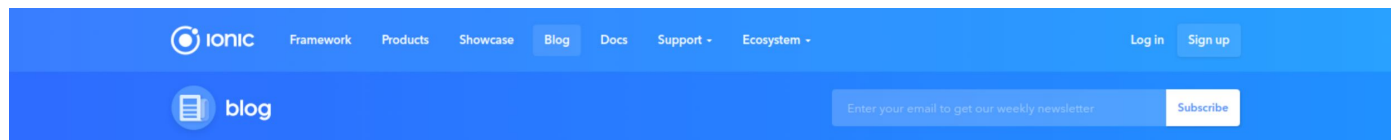
Official example built on Ionic
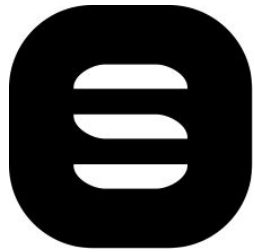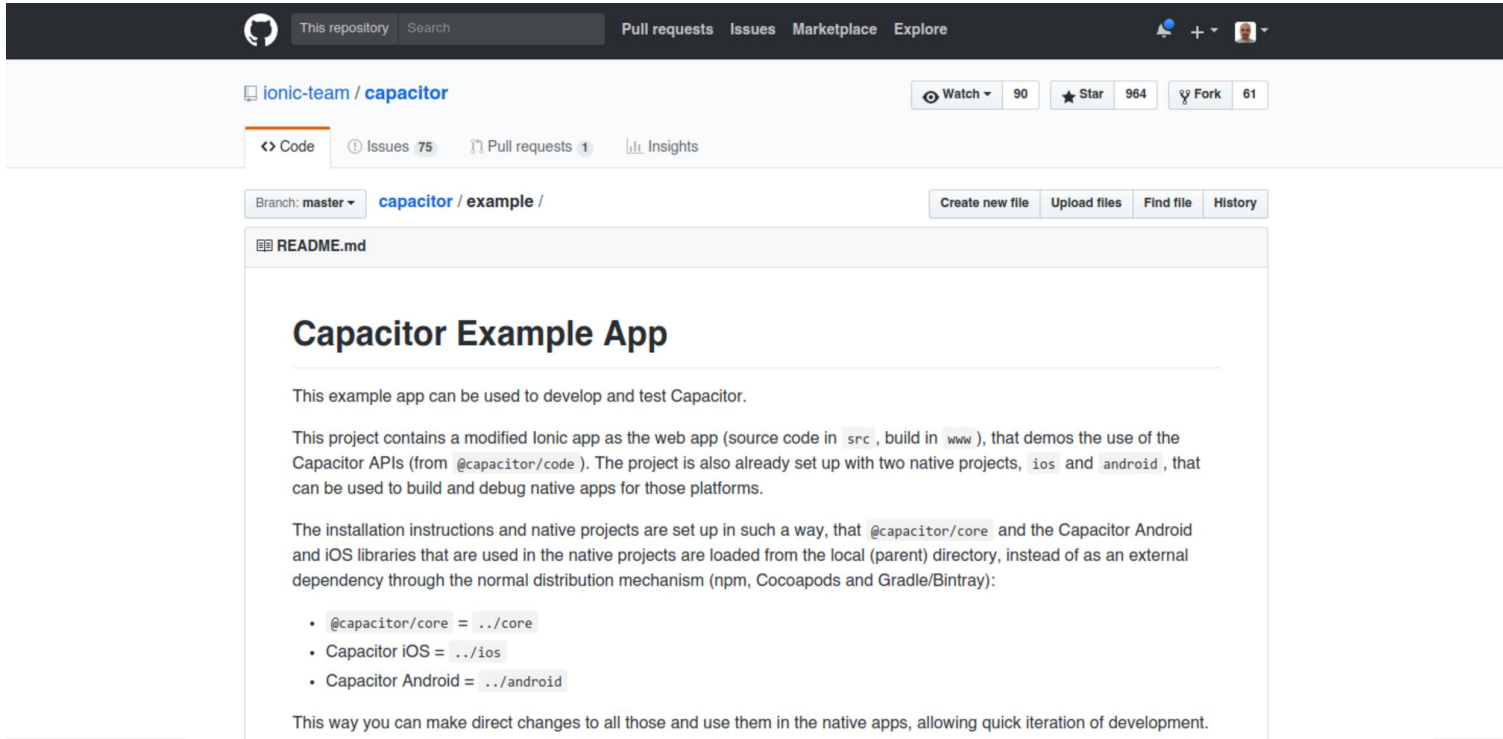
Showcasing Ionic 4

Let's try the official example

Using directly the Capacitor repository

Custom built project, not a production app

`@capacitor/core` and native libs loaded from local directory

Let's build the Android version

1. Build Capacitor Core Module

Start by building the Capacitor Core Module in `/core`:

```
cd ../core

npm install
npm run build
npm link
```

2. Build Example App

Switch back over to this example project in `/example` where you first install dependencies and link in the `@capacitor/core` you just built in the step before, then build the app and copy the build files to the correct `public` directories for both the iOS and Android example apps:

```
cd ../example

npm install
npm link @capacitor/core

npm run build
npm run copy
```

3. Build and run the native Capacitor Apps

Now that everything is in place you can build the native Capacitor Apps:

Some pre-building and building needed...

We now have an Android project
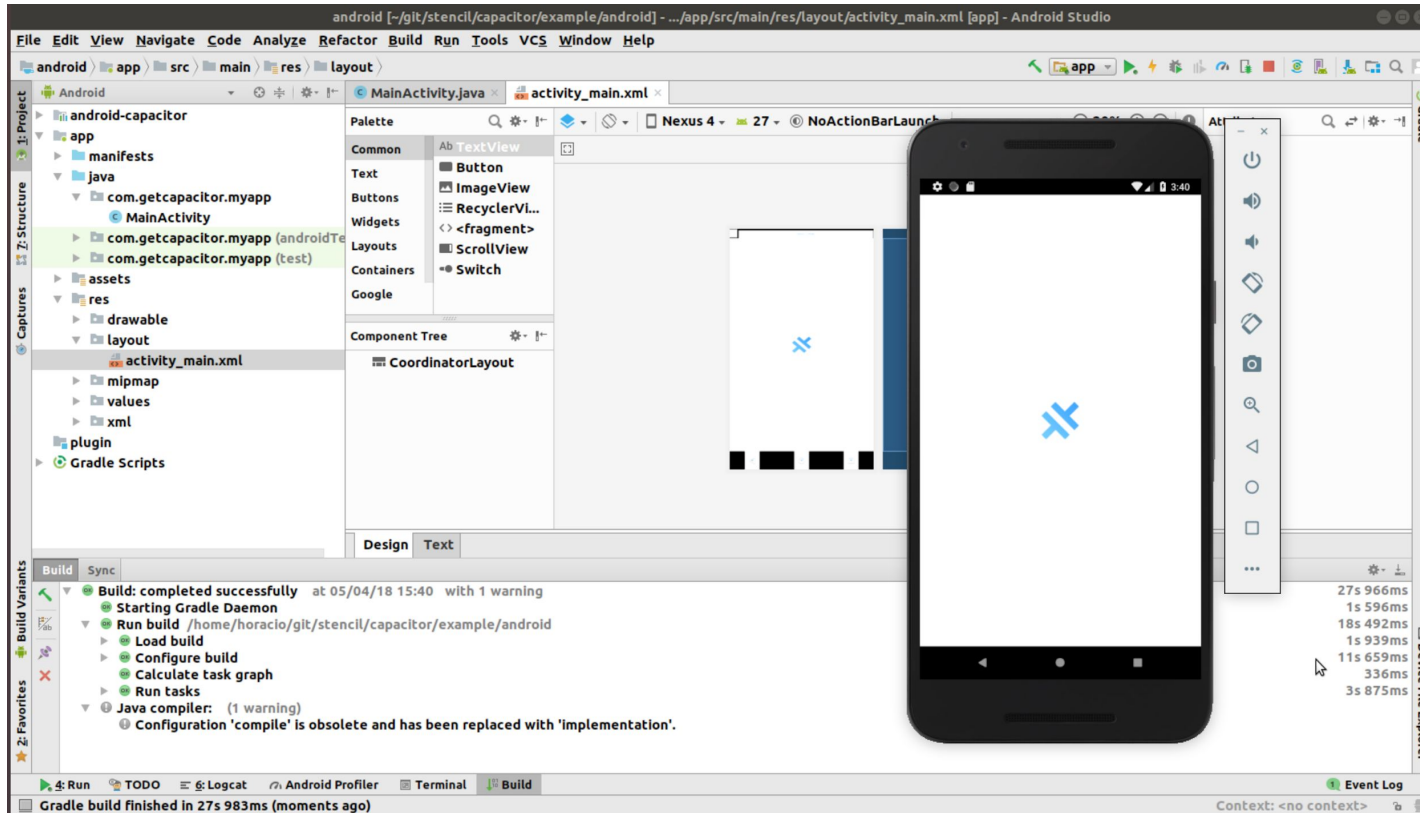
Easy and painless Android app

And in PWA mode?

Some elements haven't web implementation (yet?)

But it still works!
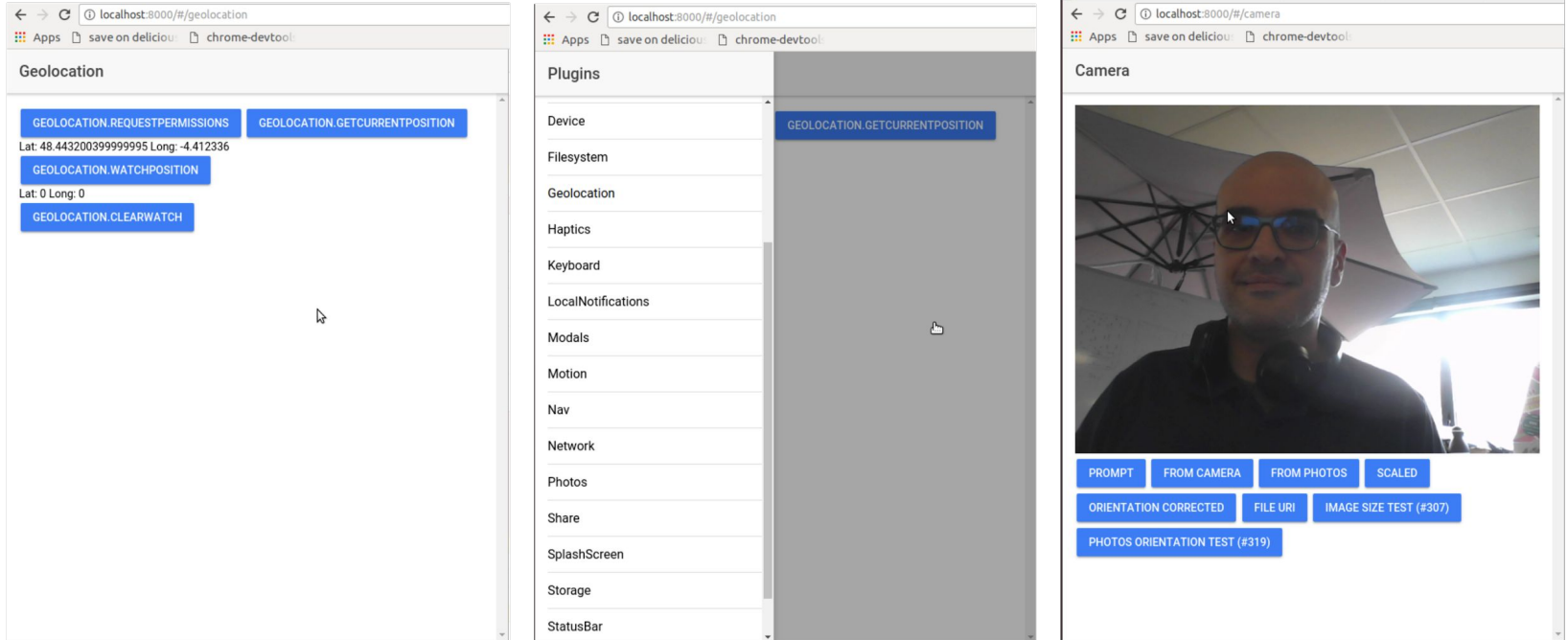
It fails gracefully for unsupported plugins

Second test: successful!

Capacitor 2 - Scepticism 0

Let's try something harder

Stencil & capacitor

Capacitor without Ionic

I want to use it Capacitor
with my own toolset

I'm a Web Components guy

And I speak a lot about Polymer

But Polymer is in a transition phase

Polymer 2.x : bower based

Polymer 3: npm based but recommended only for legacy

LitElement: the new, lightweight, blazing fast library
by Polymer team

So what to use?

Stencil, of course!

The magical, reusable web component compiler

Let's begin with a simple example

A Camera app, working well on web mode

Using standard Media Capture and Streams API

Let's charge it with Capacitor

We want the same behavior

But now in Android, iOS, Electron AND web

And does it work?
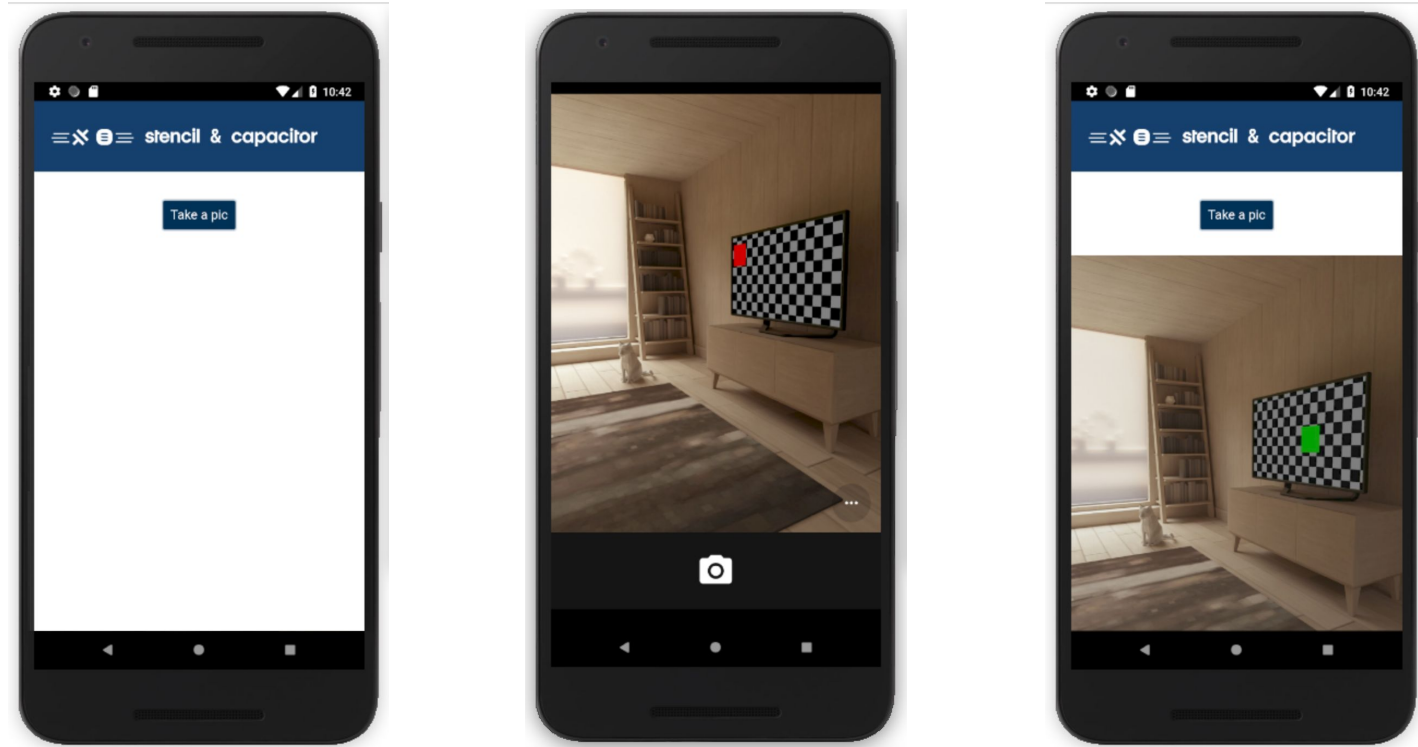
Yes it does… in native mode only!

But not in PWA mode :(
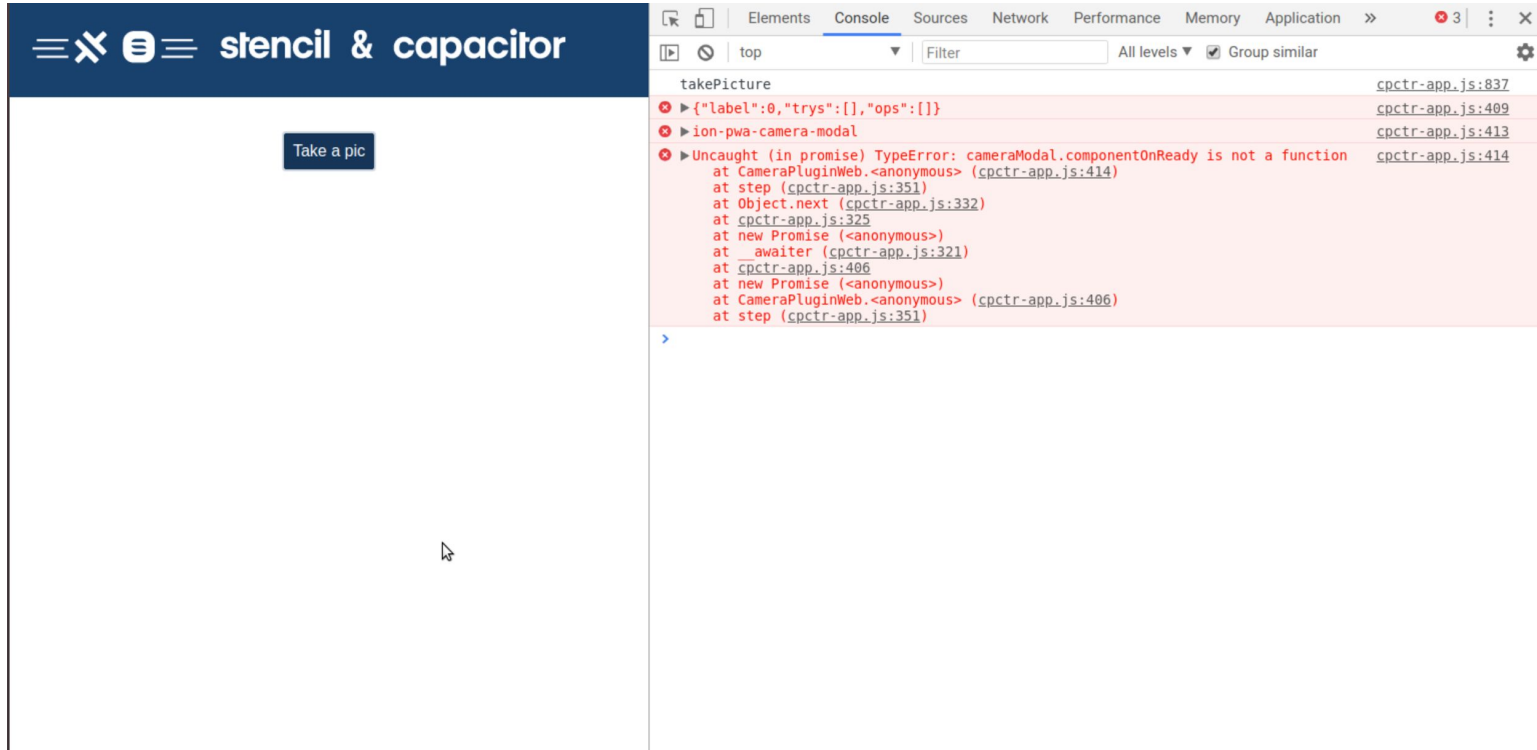
And a big question: why?

Well, let's spot the differences...

Searching on the example code

hmmmm, `@ionic/pwa-elements`, what's that?

@ionic/pwa-elements

Web-based alternatives for some Capacitor plugins

Adding @ionic/pwa-elements

Simply install them:

```
npm install @ionic/pwa-elements
```

And then import them:

```
import '@ionic/pwa-elements';
```
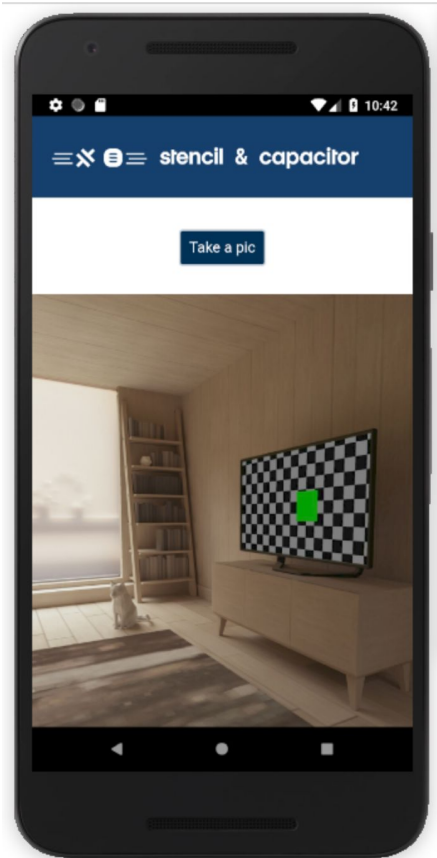
And then...

It's a kind of magic!

On Android and Web

Second test: successful!

Capacitor 3 - Scepticism 0

And in Real Life?

Because examples are examples...

Use case 1: Putting PWA into store

It simply works, easy and painless!

Use case 2: Progressive enhancing

Giving your PWA an extra P when going native

And Capacitor already works
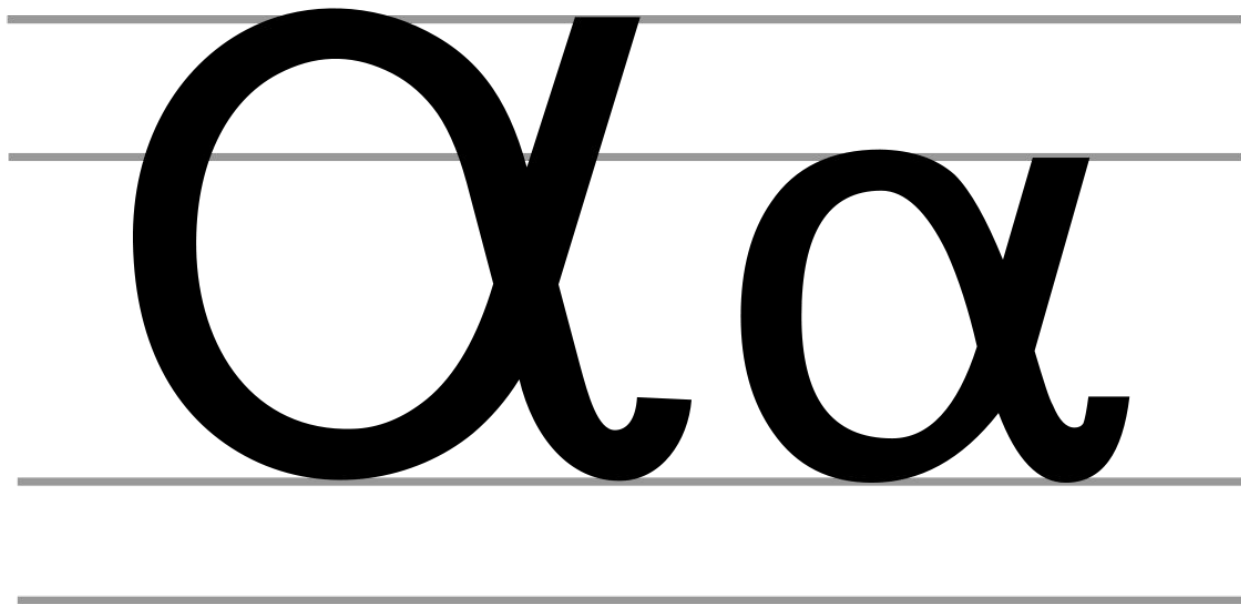

A true winner

Conclusions

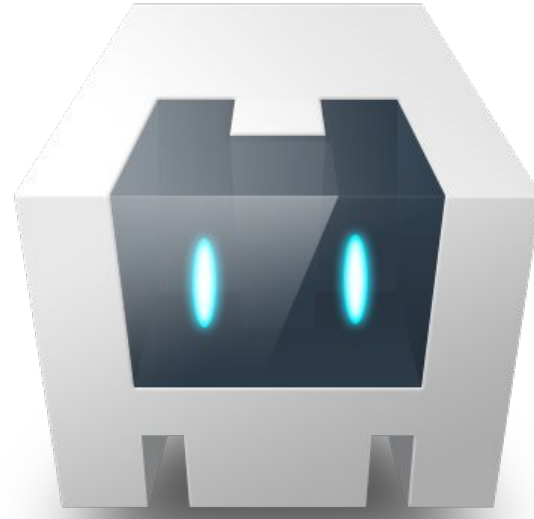## Capacitor or not?

Still fairly recent

αα

There are small glitches

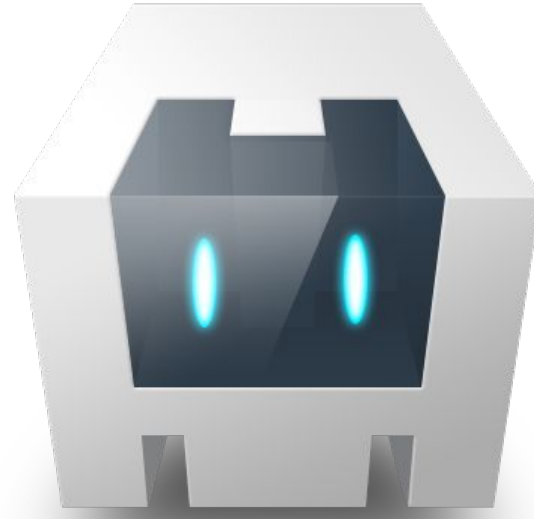Doc could be more detailed, with more examples

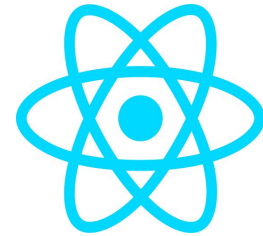# Easy to use
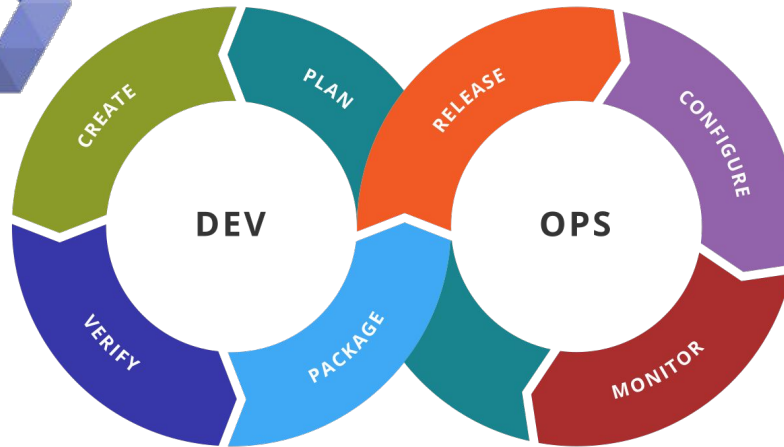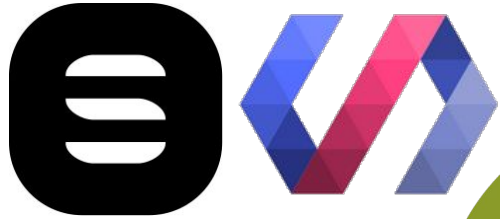

Friendlier than Cordova

Yet extensible

You can use existing Cordova plugins

Not opinionated

Easy to use in any framework
Easy to integrate in any dev toolchain

Thank you!